

云南大学数学系《运筹学通论实验》课程上机实验报告

课程名称: 运筹学通论	学期: 2015-2016 学年第二学期	成绩:
指导教师: 李建平	学生姓名: 金洋	学生学号: 20131910023
实验名称: 判断有向欧拉图		
实验编号: No.7	实验日期: 2016/6/24	实验学时: 1
学院: 数学与统计学院	专业: 信息与计算科学	年级: 2013

一、实验目的

使用 c 语言判断一幅有向图是否是欧拉图;

二、实验内容

给定一有向图 $D=(V,A)$, 判断是否存在一个有向闭合回路 C , 使 C 经过每条弧恰好一次;

三、使用环境

平台: Microsoft Visual C++ 6.0

语言: C 语言

四、算法介绍

Algorithm 判断有向欧拉图

Input $D=(V,A)$;

Output D 是否是有向欧拉图;

Begin

Step 1: 判断每个顶点是否都有 出度=入度:

若否, 输出 D 不是有向欧拉图, 结束程序;

若是, goto Step 2;

Step 2: 判断 D 是否强连通, 采用以下方法:

调用反圈法, 判断 v_s 到其余顶点是否都有路;

调用反圈法, 判断其余顶点到 v_s 是否都有路;

若否，输出 D 不是有向欧拉图，结束程序；

若是，输出 D 是有向欧拉图；

End.

五、调试过程

1. 程序代码

```
#include <stdio.h>

#include <stdlib.h>

#define MAXNUM 999999999

#define MAXVERTEX 100

int M[MAXVERTEX][MAXVERTEX],X[MAXVERTEX],n;

int inEqualsOut() {
    int outdegree[MAXVERTEX],indegree[MAXVERTEX];
    int i,j;

    /*初始化*/
    for (i=1;i<=n;i++) {
        outdegree[i]=0;
        indegree[i]=0;
    }
    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++) {
            outdegree[i]+=M[i][j];
            indegree[j]+=M[i][j];
        }
}
```

```

    for (i=1;i<=n;i++)
        if (outdegree[i]!=indegree[i]) return 0;//某顶点入度不等于出度，原图不是欧拉图；
    return 1;
}

```

/*X 是否等于 V 是:返回 1;否:返回 0*/

```

int equal() {
    //X[0]记录 x 集合中顶点个数
    return X[0]==n;
}

```

/* $\Phi^+(X)$ 是否为空? 是:返回 0;否:返回 1*/

```

int PHi_plus() {
    int i,j;
    for(i=1;i<=n;i++)
        for (j=1;j<=n;j++)
            if ((i!=j)&&(X[i])&&(X[j]==0)&&(M[i][j]<MAXNUM)) return(1);
    return(0);
}

```

```

void prim1() {
    int i,j,min,x=0,y=0;

```

```

min=MAXNUM;
for(i=1;i<=n;i++)
    if (X[i])

        for(j=1;j<=n;j++)
            if ((!X[j])&&(i!=j))
                if (M[i][j]<min) {
                    min=M[i][j];
                    x=i;
                    y=j;
                }
        if (min!=MAXNUM) {
            X[y]=1;
            X[0]++;
        }
}

/*  $\Phi^-(X)$ 是否为空? 是:返回 0;否:返回 1*/
int PHi_minus() {
    int i,j;
    for(i=1;i<=n;i++)
        for (j=1;j<=n;j++)
            if ((i!=j)&&(X[i])&&(X[j]==0)&&(M[j][i]<MAXNUM)) return(1);
    return(0);
}

```

```
}  
  
void prim2() {  
    int i,j,min,x=0,y=0;  
    min=MAXNUM;  
    for(i=1;i<=n;i++)  
        if (X[i])  
  
        for(j=1;j<=n;j++)  
            if ((!X[j])&&(i!=j))  
                if (M[j][i]<min) {  
                    min=M[j][i];  
                    x=i;  
                    y=j;  
                }  
    if (min!=MAXNUM) {  
        X[y]=1;  
        X[0]++;  
    }  
}
```

```
int stronglyConnected() {  
    int i;  
  
    //判断 V1 到其余顶点是否都有路
```

```
X[1]=1;
X[0]=1;
for (i=2;i<=n;i++) X[i]=0;
while (!equal() && PHi_plus()) prim1();
if (!equal()) return 0;

//判断其余顶点到 V1 是否都有路
X[1]=1;
X[0]=1;
for (i=2;i<=n;i++) X[i]=0;
while (!equal() && PHi_minus()) prim2();
if (!equal()) return 0;

return 1;

}

void main() {
    int i,j;
    //freopen("Euler.in", "r", stdin);
    printf("请输入 D 的顶点个数 n=");
    scanf("%d",&n);
    printf("请输入 D 的邻接矩阵:\n");
```

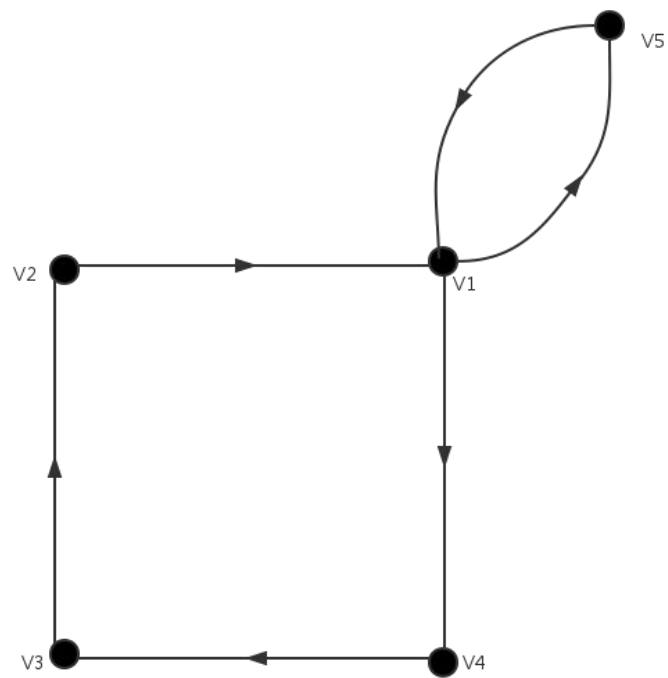
```
for (i=1;i<=n;i++)
    for (j=1;j<=n;j++) {
        scanf("%d",&M[i][j]);
        if (M[i][j]==0) M[i][j]=MAXNUM;//没有边相连，将权值设为足够大
    }

if (! inEqualsOut()) {
    printf("D 不是有向欧拉图\n");
    exit(0);
}

if (stronglyConnected()) printf("D 是有向欧拉图\n");
else printf("D 不是有向欧拉图\n");
}
```

2. 运行窗口

①

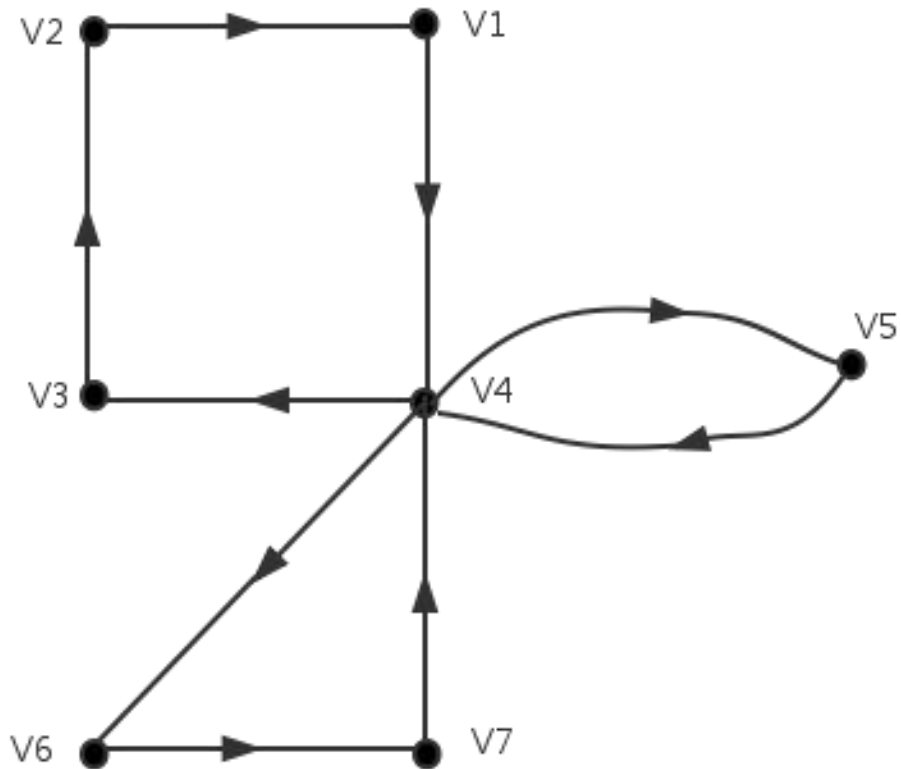


```

"F:\20131910023-金洋\程序7 (金洋-判断有向欧拉图) \Debug\Euler.exe"
请输入D的顶点个数n=5
请输入D的邻接矩阵:
0 0 0 1 1
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
1 0 0 0 0
D是有向欧拉图
Press any key to continue

```

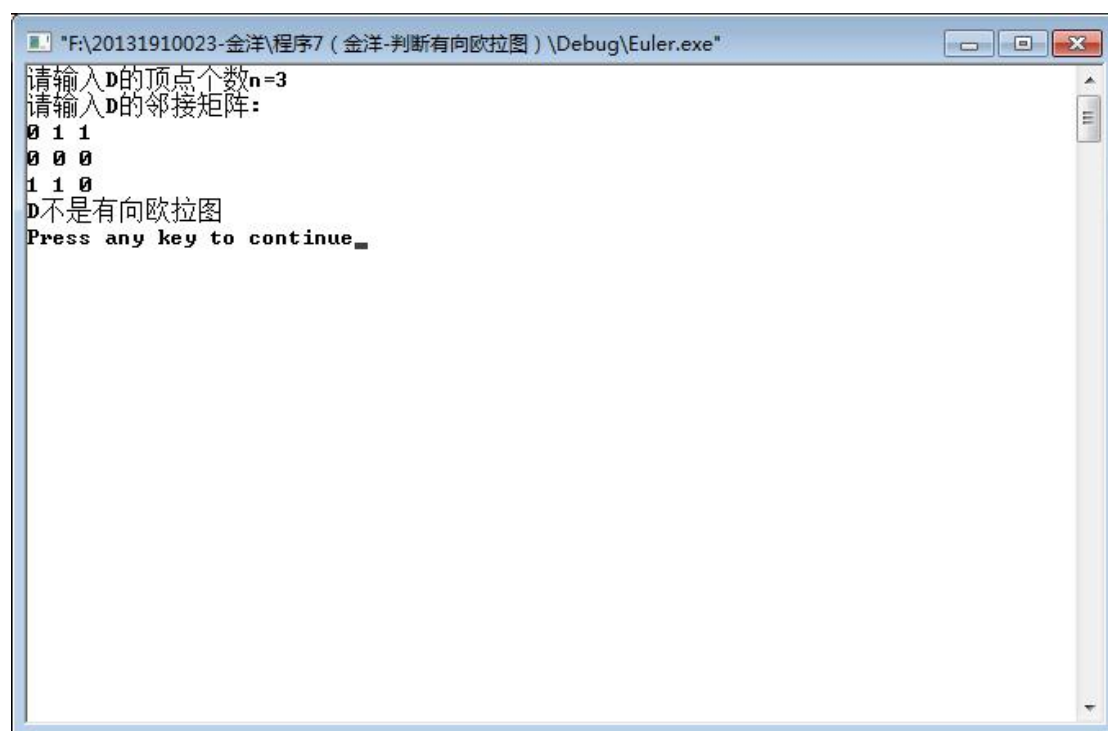
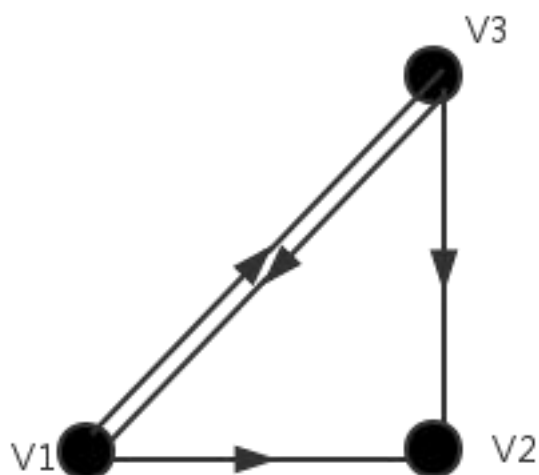
②



```

F:\20131910023-金洋\程序7 (金洋-判断有向欧拉图) \Debug\Euler.exe
请输入D的顶点个数n=?
请输入D的邻接矩阵:
0 0 0 1 0 0 0
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 1 1 0
0 0 0 1 0 0 0
0 0 0 0 0 0 1
0 0 0 1 0 0 0
D是有向欧拉图
Press any key to continue
  
```

③



六、总结

- 1.学会使用 c 语言判断一个有向图是否是强连通图，可以利用反圈法解决；
- 2.学会判断一幅有向图是否是欧拉回路。若有向图满足以下两个条件：

- ①强连通；
- ②每个顶点出度=入度；

七、参考文献

- [1] 谭浩强著,《c 程序设计》(第三版),清华大学出版社,2005.7;
- [2] 《运筹学》教程编写组,《运筹学》(第 4 版),清华大学出版社,2013.1;

八、教师评语